B: Amendments to The Claims:

What is claimed is:

1. (Currently Amended) A method for use in a computer system having an integer execution unit (FXU) having an Instruction Decode and Dispatch Unit (I-Unit) ~~and an integer execution unit (E unit)~~ of a central processor (CP) of the computer system and in which said integer execution unit contains an arithmetic and logical unit (ALU) which is capable of performing arithmetic functions including binary addition, subtraction, and logical operations such as logical and, logical or, and logical exclusive or, wherein said logical operations for operands which feed the ALU are bit maskable to select which bits of the operands participate in the logical operation, comprising the steps of:

allowing a partial instruction to be executing during the instruction dispatch cycle of the computer system by providing said integer execution unit (FXU) with a predetermination cycle pipeline stage created to accommodate a timing critical function used for execution of an instruction, and overlapping said predetermination cycle pipeline stage (E-1 stage) with a dispatch cycle of the Instruction Decode and Dispatch Unit (I-Unit) of said integer execution unit (FXU) of a central processor (CP) of the computer system before an instruction is dispatched.

2. (Currently Amended) The method according to claim 1 wherein ~~the process proceeds by dividing~~ a timing critical function used for execution of an instruction is divided into first and second partitioned processes partitioned among a virtual first and a second pipeline stage of a single pipeline, said first pipeline stage being said predetermination cycle pipeline stage and a virtual stage,

and said second pipeline stage being one where a prior instruction received has been confirmed as valid.

3. (Currently Amended) The method according to claim 2 wherein

said first partitioned process has said predetermination cycle (em1) which initiates two cycles before a first cycle of execution of said instruction when the ~~input~~ operands feed the ALU and the result of calculations performed in the ALU is put into a result register and which initiates the timing critical function for execution of an instruction to be executed.

4. (Currently Amended) The method according to claim 3, wherein

said second partitioned process, which includes said dispatch cycle, has a first predetermination cycle (e0) and a second cycle (e1) and a third cycle (PA).

5. (Original) The method according to claim 4 wherein said first predetermination cycle (e0) is initiated when an instruction is being decoded to determine what instruction it is in order to setup the controls to the computer system ALU and the second partitioned process is used for reading and loading operands into a plurality of FXU input registers of said computer system's FXU which second cycle determines whether the timing critical function of said ~~predetermined~~ predetermination cycle is valid for an instruction to be executed, and the second cycle (e1) performs the first cycle of execution of said instruction where the input operands feed the ALU and the result is put into a result register,

and wherein during the third cycle (PA) the contents of the result register are sent to an architected General Purpose Register file (GPR).

6. (Currently Amended) The method according to claim 5 wherein said third cycle is a dispatch cycle during which the contents of the result register are sent to an architected General Purpose Register file (GPR).

7. (Original) The method according to claim 5 wherein said timing critical function is a mask generation process.

8. (Original) The method according to claim 5 wherein said timing critical function determines read addresses for the GPRs of said computer system.

9. (Currently Amended) The method according to claim 5 wherein in the process allowing a partial instruction to be executing during the instruction dispatch cycle of the computer system, the three pipeline cycles of the second partitioned process used for reading and loading operands into a plurality of FXU input registers in said pipeline stages work independently so that they contain three different instructions in various states of execution.

10. (Currently Amended) The method according to claim 5 wherein the first partitioned process overlaps with the Instruction Decode and Dispatch Unit (I-unit) instruction pipeline as well as with the execution cycles of earlier in the pipeline instructions.

11. (Currently Amended) The method according to claim 5 wherein said predetermination cycle pipeline stage is used to accommodate a first part of the mask ~~generate~~ generation process, and this predetermination cycle pipeline stage (E-1 stage) is inserted before a pipeline execution stage (E0 stage).

12. (Original) The method according to claim 11, wherein said predetermination cycle pipeline stage does not increase the depth of the FXU pipeline and overlaps with the last stage or dispatch stage of the Instruction Decode and Dispatch Unit (I-Unit).

13. (Original) The method according to claim 1 wherein said predetermination cycle pipeline stage also provides cycle time relief of the first predetermination cycle (e0) execution ~~(e0)~~ stage by allowing an extra cycle to decode the instruction and form GPR read addresses which need to be launched directly from latches at the beginning of the ~~E0~~ first predetermination cycle (e0) .

14. (Original) The method according to claim 1 wherein said predetermination cycle pipeline stage is a stage of the mask ~~generator~~ generation logic which overlaps with the Instruction Decode and Dispatch Unit (I-unit) ~~instruction unit~~ as well as with the execution cycles of older (earlier in the pipeline) instructions.

15. (Original) The method according to claim 1 wherein said predetermined cycle pipeline stage is implemented as a speculated pipeline stage in which the validity of said predetermined cycle pipeline stage is not known until a following execution stage, wherein if an execution (E0) stage first becomes valid, ~~it is implied that~~ the predetermined cycle pipeline (E-1) stage is considered ~~was~~ valid on the cycle before, without impacting a subsequent dispatch stage of said I-Unit.

16. (Cancelled) .